

CLAIMS

1           1.     A method for optimizing dependencies for a set of objects comprising:  
2           automatically detecting dependencies among a set of objects, wherein each of  
3     said objects in said set includes at least one linkable file;  
4           adding said detected dependencies to a dependency list for said set of objects;  
5     and  
6           removing dependencies from said dependency list for any object that does not  
7     also have at least one file dependency.

1           2.     A method for optimizing dependencies as recited in claim 1 further  
2     comprising removing unused files from said set of objects.

1           3.     A method for optimizing dependencies for a set of objects as recited in  
2     claim 2 further comprising breaking a selected object in said set of objects into at least  
3     two smaller objects if said selected object is greater than a maximum object size.

1           4.     A method for optimizing dependencies for a set of objects as recited in  
2     claim 3 wherein said threshold maximum size is a predetermined maximum object  
3     size.

1           5.     A method for optimizing dependencies for a set of objects as recited in  
2     claim 3 further comprising making a selected file into a new object if the number of  
3     dependencies of said selected file is greater than a maximum file dependency number.

1           6.     A method for optimizing dependencies for a set of objects as recited in  
2 claim 5 wherein said maximum file dependency number is a predetermined maximum  
3 file dependency number.

1           7.     A method for optimizing dependencies for a set of objects as recited in  
2 claim 6 further comprising manually editing said dependency list.

1           8.     A method for optimizing dependencies for a set of objects as recited in  
2 claim 1 further comprising manually detecting dependencies among said set of  
3 objects, and adding said manually detected dependencies to said dependency list.

1           9.     A method for optimizing dependencies for a set of objects as recited in  
2 claim 1 wherein automatically detecting dependencies among a set of objects  
3 comprises:

4           recording dependencies to create a list of recorded dependencies during a  
5 traversal of said set of objects; and

6           analyzing said list of recorded dependencies to automatically detect  
7 dependencies.

1           10.    A method for optimizing dependencies for a set of objects as recited in  
2 claim 1 further comprising manually editing said dependency list.

1           11.    An apparatus for optimizing dependencies for a set of objects  
2 comprising:

3           means for automatically detecting dependencies among a set of objects,  
4 wherein each of said objects in said set includes at least one linkable file;

5           means for adding said detected dependencies to a dependency list for said  
6 related objects; and

7 means for removing dependencies from said dependency list for any object that  
8 does not also have at least one file dependency.

1 12. An apparatus for optimizing dependencies as recited in claim 11 further  
2 comprising means for removing unused files from said set of objects.

1 13. An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 12 further comprising means for breaking a selected object in said set of  
3 objects into at least two smaller objects if said selected object is greater than a  
4 maximum object size.

1 14. An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 13 wherein said threshold maximum size is a predetermined maximum object  
3 size.

1 15. An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 13 further comprising means for making a selected file into a new object if  
3 the number of dependencies of said selected file is greater than a maximum file  
4 dependency number.

1 16. An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 15 wherein said maximum file dependency number is a predetermined  
3 maximum file dependency number.

1 17. An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 16 further comprising means for manually editing said dependency list.

1           18.    An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 11 further comprising means for manually detecting dependencies among  
3 said set of objects, and means for adding said manually detected dependencies to said  
4 dependency list.

1           19.    An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 11 wherein said means for automatically detecting dependencies among a set  
3 of objects comprises:

4                means for recording dependencies to create a list of recorded dependencies  
5 during a traversal of said set of objects; and

6                means for analyzing said list of recorded dependencies to automatically detect  
7 dependencies.

1           20.    An apparatus for optimizing dependencies for a set of objects as recited  
2 in claim 1 further comprising means for manually editing said dependency list.

1           21.    A method for providing a tutorial comprising:

2                developing a course from an initial set of objects each including at least one  
3 linkable file, said initial set of objects being improved for at least one of transmission  
4 and storage purposes by the automatic detection of dependency information with  
5 regards to said initial set of objects and the use of said dependency information to at  
6 least one of modify an object, remove an object, split an object, and form an object to  
7 develop an improved set of objects comprising said course; and  
8                playing said course for a student.

1           22.    A method for providing a tutorial as recited in claim 21 further  
2 comprising receiving a request for said course by said student.

1           23. A method for providing a tutorial as recited in claim 22 further  
2 comprising developing a course list for presentation to said student.

1           24. A method for providing a tutorial as recited in claim 23 further  
2 comprising storing said course and said course list in a publishing database.

1           25. A method for providing a tutorial as recited in claim 21 further  
2 comprising storing multiple versions of said course in a master repository.

1           26. Computer readable media including code segments for providing a  
2 tutorial comprising:

3           a code segment for developing a course from an initial set of objects each  
4 including at least one linkable file, said initial set of objects being improved for at  
5 least one of transmission and storage purposes by the automatic detection of  
6 dependency information with regards to said initial set of objects and the use of said  
7 dependency information to at least one of modify an object, remove an object, split an  
8 object, and form an object to develop an improved set of objects comprising said  
9 course; and  
10          a code segment for playing said course for a student.

1           27. Computer readable media including code segments for providing a  
2 tutorial as recited in claim 26 further comprising a code segment for receiving a  
3 request for said course by said student.

1           28. Computer readable media including code segments for providing a  
2 tutorial as recited in claim 27 further comprising a code segment for developing a  
3 course list for presentation to said student.

1        29.    Computer readable media including code segments for providing a  
2    tutorial as recited in claim 28 further comprising a code segment for storing said  
3    course and said course list in a publishing database.

1        30.    Computer readable media including a code segment for providing a  
2    tutorial as recited in claim 26 further comprising storing multiple versions of said  
3    course in a master repository.

1        31.    An authoring environment comprising:  
2        a repository storing a set of objects comprising a course, each of said objects  
3    including at least one file;  
4        a content player coupled to said repository for receiving and playing said  
5    course, said content player including a dependency recorder which develops a  
6    dependency list with regards to said set of objects as they are played;  
7        an editor coupled to said content player and capable of editing said set of  
8    objects, said editor including a dependency analyzer which uses, at least in part, said  
9    dependency list to provide a dependency analysis for said course.

1        32.    An authoring environment as recited in claim 31 further comprising a  
2    repository explorer coupling said content player to said repository.

1        33.    An authoring environment as recited in claim 32 wherein said repository  
2    explorer includes a dependency editory which modify can an object, remove an object,  
3    split an object, and form an object to develop an improved set of objects for said  
4    course.